A purely syntactic proof of decidability for BI

Revantha Ramanayake*

Technische Universität Wien, Vienna, Austria. revantha@logic.at

Abstract. The logic of bunched implication **BI** provides a logical framework for reasoning about resource composition and systems modelling and forms the basis for an assertion language of separation logic. Reasoning about resources is critical when planning and acting in the real world as every action will generate or consume resources. The logic is obtained by freely combining propositional intuitionistic logic and multiplicative intuitionistic linear logic and yields an elegant proof theory—essentially the minimal extension of the respective sequent calculi for these logics. Most natural extensions of **BI** are undecidable. The decidability of **BI** has been shown but only via an intricate semantical argument. A purely syntactic proof of decidability has thus far proved elusive. Such a proof is obtained here by constructing a suitable measure for the size of a sequent and using this measure to bound the set of possible derivations.

1 Introduction

The logic of bunched implication **BI** [12,13] provides a logical framework expressive enough to reason about resource composition and systems modelling and forms the basis for an assertion language of separation logic [7]. The ability to reason about resources is critical when planning and acting in the real world as every action will generate or consume resources [15,8].

The logic can be defined syntactically via a sequent calculus **LBI** by taking the minimal extension of the calculus for propositional intuitionistic logic **Ip** —with the *additive* connectives $\vee, \wedge, \to$ and the constants $\top, \bot$— and multiplicative intuitionistic linear logic **MILL**—with *multiplicative* connectives $\otimes, \multimap$ and constant $\mathbf{1}$. By minimal extension we mean that **LBI** contains the structural connectives of both these calculi. In particular, the sequent calculus has a particularly elegant form where the antecedent X of a sequent $X \vdash A$ is built using two types of structural connectives: commas and semicolons (hence the name 'bunched'). The comma corresponds to $\otimes$ and the semicolon to $\wedge$. The logic **BI** is conservative over three important logics: **Ip** (which has no multiplicative connectives), **MILL** (no additive connectives) and (bounded) Distributive Commutative Full Lambek logic $\mathbf{DFL}_e$ (no intuitionistic implication $\to$).

The only proof of decidability of **BI** is the semantic proof [5] using resource tableaux. The proof there is intricate and requires the development of a large semantic framework, including objects "[to reflect] the information that can be

* Supported by the Austrian Science Fund (FWF), START project Y544.

derived from a given set of assumptions" and to built countermodels. Indeed the authors observe: "The relationships identified between resources, labels, dependency graphs, proof-search and resource semantics are central in this study [to prove decidability and finite model property]."

Certainly, [5] contains valuable results on **BI** including decidability and finite model property. Nevertheless, that proof is less accessible and difficult to check for those not familiar with the semantic methods of **BI**. The simplicity and elegance[1] of the sequent calculus for **BI** raises a question: *is it possible to prove decidability for **BI** purely syntactically?* i.e. without recourse to any semantics. We answer this question in the affirmative. The proof here is direct, uses standard sequent calculi and, in our opinion, it is more accessible and easily checkable. Moreover, since this result is syntactic, the proof calculus and decision procedure immediately yield a backward proof search procedure for **BI** and an upper-bound for the complexity of the logic.

To facilitate the presentation, we show formally in Sec. 4 that a bunch can be viewed as a 'stratified tree' of alternating commas and semicolons. The key insight (Sec. 5) is the construction of a weight function on sequents non-increasing from the conclusion to the premises of each proof rule. The weight function is technically interesting and demonstrates the scope of argument available on sequent calculi. Incidentally, **B**oolean **BI** is known to be undecidable [10].

2 Motivating the definition of the weight of a sequent

An example: decidability of intuitionistic logic. Let us begin by describing the well-known argument for proving the decidability of propositional intuitionistic logic **Ip** using its sequent calculus **LJ** [6]. Recall: B is a theorem of **Ip** iff there is a derivation of $\vdash B$ in **LJ**, where **LJ** consists of finitely many proof rules i.e. one or two premises and a conclusion (each a sequent $X \vdash A$ where X is a semicolon-separated list of formulae and A is a formula). A **LJ**-derivation is a finite tree whose nodes are labelled with sequents according to the proof rules. Thus to decide if $B \in \mathbf{Ip}$ or not, we can generate all trees which obey the rules of **LJ** with root $\vdash B$ ('candidate trees') and check if any of the candidate trees is a **LJ**-derivation (this means $B \in \mathbf{Ip}$) or not (this means $B \notin \mathbf{Ip}$).

To effectively check if the set of candidate trees contains a derivation, we want this set finite and effectively generated. Since it suffices to find a derivation of minimal height, we can immediately exclude candidate trees which contain multiple occurrences of the same sequent on a branch because such a tree would not be a derivation of minimal height. To exclude candidate trees of non-finite height, it thus suffices to show that the set of sequents that can be used to label a node is finite and effectively generated. Since **LJ** has the subformula property (following from Gentzen's *Hauptsatz*), every formula in the candidate tree must be a subformula of the root. So in a sequent $X \vdash A$ in the candidate tree, A is in the set $\mathsf{sf}(B)$ of subformulae of B, and X is a list of elements from B. However

[1] Although simplicity can be deceiving. Lambek calculus with contraction $\mathbf{FL}_c$ was recently shown to be undecidable [2]. Adding exchange ($\mathbf{FL}_{ec}$) yields decidability [9].

there are still infinitely many possible sequent labels for a node because a list can contain multiple occurrences of the same element! Now we make an observation on **Ip**: $D \wedge C \wedge C \to E \in \mathbf{Ip}$ (so $D; C; C \vdash E$ is **LJ**-derivable) iff $D \wedge C \to E \in \mathbf{Ip}$ (so $D \wedge C \vdash E$ is **LJ**-derivable). This observation tells us that for determining derivability, it suffices to consider lists X where each element occurs at most once (so semicolon has a 'contractive' property). Indeed, by further properties of **Ip** (or equivalently, **LJ**) the list is commutative as well, so we could read X simply as a set. Thus every sequent $X \vdash A$ in the candidate tree has $X \in \mathcal{P}(\mathsf{sf}(B))$ and $A \in \mathsf{sf}(\mathsf{B})$. So there are $\leq |\mathcal{P}(\mathsf{sf}(B))| \cdot |\mathsf{sf}(\mathsf{B})|$ sequents to label the tree ($\mathcal{P}$ is powerset operator).

Remark 1. The argument here does not simply eliminate the contraction rule from **LJ** (this would make the calculus incomplete for **Ip**). Instead, some contraction is built into the new left additive rules (as done in Def. 9) as it suffices for proving decidability. Aside: the contraction rule *can* be completely eliminated from **LJ** [4] replacing e.g. ($\to$l) with four specialised rules. The ensuing sequent calculus for **Ip** is contraction-free and terminates without loop check.

The sequent calculus LBI for BI. The sequent calculus **LBI** (see Fig. 1) is built from sequents of the form $X \vdash A$ where A is a formula, and X is a two-sorted list 'bunch' built from formulae and the symbols $\varnothing_m$ and $\varnothing_a$ using the list-separators comma and semicolon. The comma and semicolon lists may nest arbitrarily in the bunch. The comma (semicolon) is a structural connective corresponding to the logical connective $\otimes$ (resp. $\wedge$). Algebraically: the logical connective $\otimes$ ($\wedge$) is residuated with the linear implication $\multimap$ (resp. intuitionistic implication $\to$). As before, we will effectively generate candidate trees via backward proof search from a given bunch sequent root. The root is valid in **BI** iff one of the candidate trees is a **LBI**-derivation. It suffices to find a **LBI**-derivation of minimal height so we can exclude candidate trees which contain multiple occurrences of the same sequent on a branch. **LBI** has the subformula property so each node in a candidate tree is labelled by a bunch sequent built using subformulae from the root.

The difficulty with **LBI** arises from the fact that although the comma and semicolon are associative and commutative ($\varnothing_m$ and $\varnothing_a$ are the respective identity elements), and semicolon has (nice) contractive and weakening properties, the comma does not. So $p, p \vdash p \otimes p$ is **LBI**-derivable but $p \vdash p \otimes p$ and $p, p, q \vdash p \otimes p$ are not. Certainly we can restrict our attention to simplified bunches in **LBI** such that semicolon-separated lists do not contain multiple occurrences of the same element and $\varnothing_m, X \mapsto X$ and $\varnothing_a; X \mapsto X$ (such bunch$^{*\mathsf{r}}$ structures are formally related to bunches in Sec. 4). We observe that there is no rule in **LBI** which can 'increase' the length of comma-separated lists except via duplication using semicolon (but this is the contractive effect we have already abstracted away by moving to bunch$^{*\mathsf{r}}$!). So the challenge is to bound the length of comma-separated lists (i.e. the number of contiguous commas in a bunch).

We want to measure the size of a sequent such that (i) the size is nonincreasing from conclusion to premise and (ii) the size bounds the length of comma-

separated lists, and implies that number of different sequents of size less than a fixed value is finite and computable. The obvious candidate for the size of a sequent $X \vdash A$ is $|X|+|A|$ where $|\cdot|$ extends the standard definition of the size of a formula to bunches. In particular, $|U, V| = |U|+|V|+1$ and $|U; V| = |U|+|V|+1$. However, then the contraction rule (ctr) in **LBI** would violate objective (i). The point is that the premise would have greater size than the conclusion because $X; X$ would have greater size than X. Solution: define $|U; V|$ as $\max(|U|, |V|)$. *Count commas, take maximum over semicolon.* Now consider the following:

$$\frac{Y \vdash C \qquad D \vdash A}{Y; C \to D \vdash A}\ (\to\mathrm{l}) \qquad \frac{C; C \vdash A}{C \vdash A}\ (\mathrm{ctr}) \qquad \frac{C \multimap D, C \vdash D}{C \multimap D \vdash C \multimap D}\ (\multimap\mathrm{r})$$

(Above left) in the conclusion we have $\max(|Y|, |C \to D|) + |A|$ but the size of the left premise $Y \vdash C$ is $|Y| + |C|$. So if $|Y|$ and $|C|$ are large relative to the other variables, then $|Y| + |C| > |Y| + |A|$ and $|Y| + |C| > |C \to D| + |A|$ and thus the premise would have greater size than the conclusion. In response we must somehow preemptively *take into account* $|U| + |B|$ *for every substructure* $U; B$ *when* B *is a formula.*

However, if we count the substructure $U; B$ as $|U| + |B|$ then (ctr) rule again causes difficulties. (Above centre) the premise has size $\approx |C| + |C|$ compared to the conclusion size $\approx |C|$ when $|C| \gg |A|$. In response we must preemptively *take into account* $|C| + |C|$ *for every formula* C occurring in the sequent. In this way we finally obtain a measure achieving objective (i).

As an aside, (above right) indicates why $\max(|X|, |A|)$ is an inadequate measure for the size of a sequent $X \vdash A$. Under this measure the premise would have size $2|C| + |D| + 2$ which is greater than $|C| + |D| + 1$.

Summary: the set $\mathsf{CP}(X \vdash A)$ of *critical pairs* (Def. 11) of $X \vdash A$ consists of:

1. $\{X\}\{A\}$
2. $\{B\}\{B\}$ for every formula B in $X \vdash A$
3. $\{U\}\{B\}$ for $U; B$ in X.

The size of a critical pair $\{U\}\{B\}$ is defined as $|U| + |B|$. The *weight* of $X \vdash A$ is then the maximum of the sizes of the critical pairs in $\mathsf{CP}(X \vdash A)$ (Def. 13). Notice that the maximum length of a comma-separated list in X cannot be greater than the weight of $X \vdash A$ so we have achieved objective (ii).

3 Preliminaries

We assume a countable infinite set $\mathcal{V}$ of propositional variables.

The language of *intuitionistic* logic **Ip** consists of the logical constants $\bot$ and $\top$, the unary logical connectives $\wedge$ and $\vee$ and the binary logical connective $\to$. A formula in intuitionistic logic is built from the propositional variables and logical constants using the connectives.

The language of *multiplicative intuitionistic linear logic* **MILL** has the logical constant $\mathbf{1}$ and the binary logical connectives $\otimes$ and $\multimap$. A formula in **MILL** is built from the propositional variables and logical constants using the connectives.

The logic of bunched implications **BI** is obtained as the free combination of **Ip** and **MILL**. By this we mean the logic whose logical connectives and presentation are the union of the presentations for **Ip** and **MILL**.

A *formula* of **BI** is a finite term from the following grammar.

$$A := p \in \mathcal{V} \mid \top \mid \bot \mid \mathbf{1} \mid (A \vee A) \mid (A \wedge A) \mid (A \to A) \mid (A \otimes A) \mid (A \multimap A)$$

The set of formulae is denoted Fm. For new symbols $\varnothing_m$ and $\varnothing_a$ define $\mathsf{Fm}^{\varnothing} := \mathsf{Fm} \cup \{\varnothing_m, \varnothing_a\}$. A *bunch* is a finite term from the following grammar:

$$X := A \in \mathsf{Fm}^{\varnothing} \mid \varnothing_a \mid \varnothing_m \mid (X, X) \mid (X; X)$$

Definition 1 (sequent, sequent rule, sequent calculus). *A* sequent *(denoted $X \vdash A$) is an ordered pair where X is a bunch and A is a formula. The structure X is called the* antecedent *of the sequent.*

A sequent rule *is typically written as follows for $n \geq 0$.*

$$\frac{X_1 \vdash A_1 \qquad \dots \qquad X_n \vdash A_n}{X_0 \vdash A_0}$$

The sequents above the line are called the premises *and the sequent below the line is called the* conclusion*. A rule with no premises is called an* initial sequent*.*

A sequent calculus consists of a set of sequent rules.

Definition 2 (derivation). *A derivation in a sequent calculus is defined recursively in the usual way as an initial sequent or the object obtained by applying some sequent rule to a smaller derivation.*

The *height of a derivation* is defined in the usual way as the number of sequents along its longest branch.

Notation. Any formula built using binary connectives can be viewed in a natural way as an ordered binary tree. We write U, V $(U; V)$ to mean a tree with root node comma (resp. semicolon) and children U and V. We write $\Gamma = \Gamma[A]$ $(A \in \mathsf{Fm}^{\varnothing})$ to indicate a specific occurrence of A in Γ. Also $\Gamma[U, V]$ indicates that Γ contains a comma node with two children U and V. Similarly $\Gamma[U; V]$ indicates that Γ contains a semicolon node with two children U and V. Later we extend this notation to non-binary trees (see after Def. 6). The symbol $=$ is used to denote syntactic equality.

Definition 3 (LBI). *The sequent calculus **LBI** consists of the rules in Fig.1 where the antecedent of every sequent is read as a* bunch*.*

LBI is identical to the original calculus [13,5] except (1) for technical convenience we repeat in the premise the principal formula of additive left introduction rules—equivalence with the original rules is immediate due to (weak) and (ctr)—and (2) we explicitly present the associativity, exchange and identity rules for comma and semicolon. In contrast, the original calculus uses the following rule:

(a) Initial sequents, logical constants and proper structural rules:

$$\frac{}{C \vdash C}\ C \in \mathsf{Fm} \qquad \frac{}{\varnothing_m \vdash \mathbf{1}}\ (\mathbf{1}\mathrm{r}) \qquad \frac{}{\Gamma[\bot] \vdash C}\ (\bot\mathrm{l}) \qquad \frac{}{X \vdash \top}\ (\top\mathrm{r})$$

$$\frac{\Gamma[\varnothing_m] \vdash A}{\Gamma[\mathbf{1}] \vdash A}\ (\mathbf{1}\mathrm{l}) \qquad \frac{\Gamma[X] \vdash A}{\Gamma[X;Y] \vdash A}\ (\text{weak}) \qquad \frac{\Gamma[X;X] \vdash A}{\Gamma[X] \vdash A}\ (\text{ctr})$$

(b) Rules simulating Pym's (E) rule:

$$\frac{\Gamma[(X,Y),Z] \vdash A}{\Gamma[X,(Y,Z)] \vdash A}\ (\text{as-c}) \qquad \frac{\Gamma[X,Y] \vdash A}{\Gamma[Y,X] \vdash A}\ (\text{ex-c}) \qquad \frac{\Gamma[X] \vdash A}{\Gamma[\varnothing_m, X] \vdash A}\ (\varnothing_m\mathrm{l})$$

$$\frac{\Gamma[(X;Y);Z] \vdash A}{\Gamma[X;(Y;Z)] \vdash A}\ (\text{as-sc}) \qquad \frac{\Gamma[X;Y] \vdash A}{\Gamma[Y;X] \vdash A}\ (\text{ex-sc}) \qquad \frac{\Gamma[X] \vdash A}{\Gamma[\varnothing_a; X] \vdash A}\ (\varnothing_a\mathrm{l})$$

(c) Additives:

$$\frac{Y;C \to D \vdash C \qquad \Gamma[D;C \to D] \vdash A}{\Gamma[Y;C \to D] \vdash A}\ (\to l) \qquad \frac{\Gamma[C;C \vee D] \vdash A \qquad \Gamma[D;C \vee D] \vdash A}{\Gamma[C \vee D] \vdash A}\ (\vee\mathrm{l})$$

$$\frac{X;C \vdash D}{X \vdash C \to D}\ (\to r) \qquad \frac{\Gamma[C_i;C_1 \wedge C_2] \vdash A}{\Gamma[C_1 \wedge C_2] \vdash A}\ (\wedge\mathrm{l}) \qquad \frac{X \vdash C \qquad X \vdash D}{X \vdash C \wedge D}\ (\wedge\mathrm{r}) \qquad \frac{\Gamma \vdash C_i}{\Gamma \vdash C_1 \vee C_2}\ (\vee\mathrm{r})$$

(d) Multiplicatives:

$$\frac{Y \vdash C \qquad \Gamma[D] \vdash A}{\Gamma[Y, C \mathrel{-\!\!\circ} D] \vdash A}\ (\mathrel{-\!\!\circ} l) \qquad \frac{X, C \vdash D}{X \vdash C \mathrel{-\!\!\circ} D}\ (\mathrel{-\!\!\circ} r)$$

$$\frac{\Gamma[C,D] \vdash A}{\Gamma[C \otimes D] \vdash A}\ (\otimes\mathrm{l}) \qquad \frac{X \vdash C \qquad Y \vdash D}{X, Y \vdash C \otimes D}\ (\otimes\mathrm{r})$$

Fig. 1. Rule format for **LBI**, **LBI*** and **LBI**$^{*\mathsf{r}}$ calculi. The double lines are notation to succinctly write two rules, via the upwards and downward direction. For **LBI**, each antecedent is read as a bunch. For **LBI***, each antecedent is read as a bunch* and the rules (as-c), (as-sc), (ex-c) and (ex-sc) are deleted. For **LBI**$^{*\mathsf{r}}$, (i) each antecedent is read as a bunch$^{*\mathsf{r}}$ (ii) all the rules in (b) and the (ctr) rule are deleted, and (iii) the rules ($\mathrel{-\!\!\circ}'$r), ($\to'$r), ($\otimes'$r), ($\otimes''$r) and ($\otimes'''$r) are added (see Def. 9).

$$\frac{X \vdash A}{Y \vdash A}\ (\mathrm{E})\ X \equiv Y$$

The equivalence relation $\equiv$ is specified as (i) the commutative monoid equations for $\varnothing_m$ and ",", (ii) the commutative monoid equations for $\varnothing_a$ and ";" and (iii) congruence: if $X \equiv Y$ then $\Gamma[X] \equiv \Gamma[Y]$.

Following the standard terminology, the occurrences of the formulae in the premise are called the *active formula(e)* of the rule. Meanwhile the formula in the conclusion is called the *principal formula* of the rule.

Definition 4 (interpretation of a structure). *The* interpretation X^I *of a structure X is the formula obtained by reading each comma as $\otimes$, semicolon as $\wedge$, $\varnothing_m$ as $\mathbf{1}$ and $\varnothing_a$ as $\top$.*

Several different semantics for **BI** have been proposed [13,14,5], including the important resource semantics. Since the focus of this paper is on the syntax, we

refer the reader to the literature for the details. The algebraic semantics are given by the class of Heyting algebras which carry an additional ordered commutative monoid structure with binary operation $\otimes$, identity $\mathbf{1}$ and linear implication $\multimap$ such that $x \otimes y \leq z$ iff $x \leq y \multimap z$ ($\leq$ is the Heyting lattice order).

Theorem 1 ([13]). *$X \vdash A$ is **LBI**-derivable iff $X^I \leq A$ holds on all **BI**-algebras under all assignments.*

4 A normal form for bunches

Definition 5 (multiplicative, additive bunch). *A bunch is* multiplicative *(*additive*) if its headconnective is* comma *(resp.* semicolon*).*

A bunch can be viewed as an ordered binary tree whose vertex set is $\mathsf{Fm}^{\varnothing} \cup \{\mathsf{comma}, \mathsf{semicolon}\}$ such that every leaf is in $\mathsf{Fm}^{\varnothing}$ and every interior node is in $\{\mathsf{comma}, \mathsf{semicolon}\}$. Then a *canonical bunch* with root comma (semicolon) is a binary tree whose left branch is either in $\mathsf{Fm}^{\varnothing}$ or a canonical bunch with root semicolon (resp. comma), and whose right branch is canonical.

Every bunch is $\equiv$-equivalent to a canonical bunch [1]. To facilitate our argument here we will now introduce an unordered tree data structure which can be obtained from a canonical bunch by identifying each maximal sequence of contiguous commas (semicolons) along the right branches with a single comma (semicolon), obtaining a 'stratified' tree of alternating commas and semicolons. Each interior node will thus have 2 or more children.

Definition 6 (bunch*). *A* bunch* *is a finite object defined recursively as*

1. *A single node from $\mathsf{Fm}^{\varnothing}$ and no edges.*
2. *A node* comma *with 2 or more children such that no child is a multiplicative bunch*.*
3. *A node* semicolon *with 2 or more children such that no child is an additive bunch*.*

Notation. $U_1, \ldots, U_{n+2}$ ($U_1; \ldots; U_{n+2}$) for $n \geq 0$ denotes a tree with root node comma (resp. semicolon) and children U_i. Also $\Gamma[U_1, \ldots, U_{n+2}]$ denotes that Γ contains a node comma with children U_i. Note that this comma may also have other children that are not explicitly named here (previously we considered binary trees so this was not a possibility). Define $\Gamma[U_1; \ldots; U_{n+2}]$ analogously.

The following program takes as input a bunch X and returns X^*.

```
1  input X
2  if X ∈ Fm^∅ then X* := X
3  else if X = U,V then obtain U* and V*
4   case U* and V*
5          U1,...,Un+2 and V1,...,Vm+2 :  X* := U1,...,Un+2,V1,...,Vm+2
6          U1,...,Un+2 and V1;...;Vm+2 :  X* := U1,...,Un+2,V*
7          U1;...;Un+2 and V1,...,Vm+2 :  X* := U*,V1,...,Vm+2
```

```
 8          U_1;...;U_{n+2} and V_1;...;V_{m+2}:  X* := U*, V*
 9  else if X = U;V then obtain U* and V*
10   case U* and V*
11          U_1,...,U_{n+2} and V_1,...,V_{m+2}:  X* := U*; V*
12          U_1,...,U_{n+2} and V_1;...;V_{m+2}:  X* := U*; V_1;...;V_{m+2}
13          U_1;...;U_{n+2} and V_1,...,V_{m+2}:  X* := U_1;...;U_{n+2}; V*
14          U_1;...;U_{n+2} and V_1;...;V_{m+2}:  X* := U_1;...;U_{n+2}; V_1;...;V_{m+2}
15  return X*
```

Lemma 1. *Let X be a bunch. Then X^* is a bunch*.*

Proof. Induction on the size of X. Base case: $X \in \mathsf{Fm}^{\varnothing}$ so X is already a bunch*.

In the inductive case, suppose that $X = U, V$ (the case of $U; V$ is similar). Then the induction hypothesis tells us that U^* and V^* are bunch*. By inspection, lines 5–8 of the program ensure that X^* is a bunch*. Case $U; V$ is analogous. □

Definition 7 (LBI*). *The sequent calculus **LBI*** consists of the rules in Fig.1 minus (as-c), (as-sc), (ex-c) and (ex-sc) where the antecedent of every sequent is read as a* bunch*.

By "where the antecedent of every sequent is read as a *bunch**" we mean that the rule instances of **LBI*** are precisely those obtained by applying *bunch** to the antecedents of the premise(s) and conclusions of **LBI** rule instances.

Example 1. Below left we give the derivation of $p \otimes (q \otimes r) \vdash (p \otimes r) \otimes q$ in **LBI**. The **LBI*** calculus uses the bunch* data structure in place of the bunch data structure in **LBI**. As a result, in the derivation in **LBI*** below right, the details of the exchange and associative rules are abstracted away.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{p \vdash p \qquad r \vdash r}{p, r \vdash p \otimes r}\,(\otimes\mathrm{r}) \qquad q \vdash q}{(p, r), q \vdash (p \otimes r) \otimes q}\,(\otimes\mathrm{r})}{p, (r, q) \vdash (p \otimes r) \otimes q}\,(\text{as-c})}{p, (q, r) \vdash (p \otimes r) \otimes q}\,(\text{ex-c})}{p, (q \otimes r) \vdash (p \otimes r) \otimes q}\,(\otimes\mathrm{l})}{p \otimes (q \otimes r) \vdash (p \otimes r) \otimes q}\,(\otimes\mathrm{l})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{p \vdash p \qquad r \vdash r}{p, r \vdash p \otimes r}\,(\otimes\mathrm{r}) \qquad q \vdash q}{p, q, r \vdash (p \otimes r) \otimes q}\,(\otimes\mathrm{r})}{p, q \otimes r \vdash (p \otimes r) \otimes q}\,(\otimes\mathrm{l})}{p \otimes (q \otimes r) \vdash (p \otimes r) \otimes q}\,(\otimes\mathrm{l})$$

□

As we would expect, **LBI** and **LBI*** are equally expressive:

Lemma 2. *(i) If $X \vdash A$ is **LBI**-derivable then $X^* \vdash A$ is **LBI***-derivable. (ii) If $Y \vdash A$ is **LBI***-derivable and $X^* = Y$, then $X \vdash A$ is **LBI**-derivable.*

Proof. (i) Induction on the height of the derivation of $X \vdash A$. Consider the last rule ρ of the derivation and apply the induction hypothesis to its premise(s). If the last rule was (as-c), (as-sc), (ex-c) or (ex-sc) we have already obtained the required derivation. Otherwise reapply ρ (this time in **LBI***).

(ii) Induction on the height of the derivation of $Y \vdash A$. Let $Y' \vdash A'$ denote the premise of the last rule ρ of the derivation (the argument is similar when ρ is binary). Obtain the bunch X_1 (ordered binary tree) from the bunch* Y' (unordered tree) by interpreting the commas and semicolons with left-associative precedence i.e. $U_1, \ldots, U_{n+2}$ becomes $((\ldots((U_1, U_2), U_3), \ldots), U_{n+2})$. Then $X_1^* = Y'$ and $Y' \vdash A'$ is $\mathbf{LBI}^*$-derivable so the induction hypothesis yields that $X_1 \vdash A'$ is $\mathbf{LBI}$-derivable. Note that X_1 is identical to Y' if the parentheses are ignored—in particular, any active formulae of ρ in Y' also appear in X_1—use (as-c), (as-sc), (ex-c) and (ex-sc) as required in order to apply ρ (this time in $\mathbf{LBI}$) with conclusion $X_0 \vdash A$. Now X_0 and X differ only in parenthetical ordering, so $X \vdash A$ is derivable from $X_0 \vdash A$ using (as-c), (as-sc), (ex-c) and (ex-sc). □

Now we introduce a more nuanced notion of a bunch* which removes comma-separated $\varnothing_m$, semicolon-separated $\varnothing_a$ and semicolon-separated duplicates.

Definition 8 (bunch$^{*\mathsf{r}}$). *A* bunch$^{*\mathsf{r}}$ *is a finite object defined recursively as*

1. *A single node from* $\mathsf{Fm}^{\varnothing}$ *and no edges.*
2. *A node* comma *with* 2 *or more children such that no child is a multiplicative bunch*$^{*\mathsf{r}}$*. Also* no child is $\varnothing_m$.
3. *A node* semicolon *with* 2 *or more children such that no child is an additive bunch*$^{*\mathsf{r}}$*. Also* no child is $\varnothing_a$ and no two children are identical.

A *bunch*$^{*\mathsf{r}}$ *sequent* is a sequent whose antecedent is a bunch$^{*\mathsf{r}}$. Clearly a bunch$^{*\mathsf{r}}$ is a bunch* but the other direction does not hold in general. To be concrete, let us define an explicit function r on bunch* to achieve this transformation. For $A \in \mathsf{Fm}^{\varnothing}$ define $A^{\mathsf{r}} = A$. Otherwise:

$$(U_1, \ldots, U_{n+2})^{\mathsf{r}} = \begin{cases} \varnothing_m & \text{if } U_i^{\mathsf{r}} = \varnothing_m \text{ for every } i \\ U^{\mathsf{r}}_{s_1}, \ldots, U^{\mathsf{r}}_{s_{t+2}} & (s_1, \ldots, s_{t+2}) \text{ subseq. of } (1, \ldots, n+2) \\ & j \in \{s_1, \ldots, s_{t+2}\} \text{ iff } U_j^{\mathsf{r}} \neq \varnothing_m \end{cases}$$

$$(U_1; \ldots; U_{n+2})^{\mathsf{r}} = \begin{cases} \varnothing_a & \text{if } U_i^{\mathsf{r}} = \varnothing_a \text{ for every } i \\ U^{\mathsf{r}}_{s_1}; \ldots; U^{\mathsf{r}}_{s_{t+2}} & (s_1, \ldots, s_{t+2}) \text{ subseq. of } (1, \ldots, n+2) \\ & j \in \{s_1, \ldots, s_{t+2}\} \text{ iff } U_j^{\mathsf{r}} \neq \varnothing_a \\ & \text{For } j, k \in \{s_1, \ldots, s_{t+2}\}\text{: } U_j^{\mathsf{r}} = U_k^{\mathsf{r}} \text{ iff } j = k \end{cases}$$

Note. We will implicitly use the observation that $(X^{\mathsf{r}}; Y)^{\mathsf{r}}$ is identical to $(X, Y)^{\mathsf{r}}$.

Lemma 3. *Every bunch** X *can be effectively written as a bunch*$^{*\mathsf{r}}$ $X^{*\mathsf{r}}$.

Lemma 4. *Let Y be a bunch*$^{*\mathsf{r}}$ *and $X^{\mathsf{r}} = Y$ for some bunch** X*. Then there is a* $\boldsymbol{LBI}^*$*-derivation of $\Gamma[X] \vdash A$ from $\Gamma[Y] \vdash A$ for any $\Gamma[\,]$ and $A \in \mathsf{Fm}$.*

Proof. Let $\#\mathsf{r}(X)$ denote the number of recursive calls of r (including the original function call) that witness $X^{\mathsf{r}} = Y$. Argue by induction on $N(X)$. If $\#\mathsf{r}(X) = 1$ (base case) then X is already a bunch$^{*\mathsf{r}}$ or $X = \varnothing_m, \ldots, \varnothing_m$ or $X = \varnothing_a, \ldots, \varnothing_a$. In the first case we already have the required derivation, in the remaining two cases we proceed in $\mathbf{LBI}^*$ as follows.

$$\frac{\Gamma[\varnothing_m] \vdash A}{\Gamma[\varnothing_m, \ldots, \varnothing_m] \vdash A}\,(\varnothing_m\mathrm{l}) \qquad \frac{\Gamma[\varnothing_a] \vdash A}{\Gamma[\varnothing_a; \ldots; \varnothing_a] \vdash A}\,(\varnothing_a\mathrm{l})$$

Now suppose that $\#\mathsf{r}(X) = k+1$. From the definition of r we have (i) $X = U_1, \ldots, U_{n+2}$ and $X^{\mathsf{r}} = U^{\mathsf{r}}_{s_1}, \ldots, U^{\mathsf{r}}_{s_{t+2}}$ or (ii) $X = U_1; \ldots; U_{n+2}$ and $X^{\mathsf{r}} = U^{\mathsf{r}}_{s_1}; \ldots; U^{\mathsf{r}}_{s_{t+2}}$. Noting that $\#\mathsf{r}(U_{s_i}) \leq k$ for every i, we proceed in $\mathbf{LBI}^*$:

$$\dfrac{\dfrac{\Gamma[U^{\mathsf{r}}_{s_1}, \ldots, U^{\mathsf{r}}_{s_{t+2}}] \vdash A}{\Gamma[U_{s_1}, \ldots, U_{s_{t+2}}] \vdash A}\,\text{IH}}{\Gamma[U_1, \ldots, U_{n+2} \vdash A}\,(\varnothing_m\mathrm{l})\text{ rules} \qquad \dfrac{\dfrac{\Gamma[U^{\mathsf{r}}_{s_1}; \ldots; U^{\mathsf{r}}_{s_{t+2}}] \vdash A}{\Gamma[U_{s_1}; \ldots; U_{s_{t+2}}] \vdash A}\,\text{IH}}{\Gamma[U_1; \ldots; U_{n+2} \vdash A}\,(\varnothing_a\mathrm{l})\text{ and (weak) rules}}$$

□

Definition 9 ($\mathbf{LBI}^{*\mathsf{r}}$). *The sequent calculus* $\boldsymbol{LBI}^{*\mathsf{r}}$ *consists of the rules in Fig. 1 minus the rules in (b) and the (ctr) rule, where the antecedent of every sequent is read as a* bunch$^{*\mathsf{r}}$*. Add rules* $(\multimap' r)$*,* $(\to' r)$*,* $(\otimes' r)$*,* $(\otimes'' r)$*,* $(\otimes''' r)$*:*

$$\frac{C \vdash D}{\varnothing_m \vdash C \multimap D}\,(\multimap' r) \qquad \frac{C \vdash D}{\varnothing_a \vdash C \to D}\,(\to' r) \qquad \frac{\varnothing_m \vdash C \quad \varnothing_m \vdash D}{\varnothing_m \vdash C \otimes D}\,(\otimes' r)$$

$$\frac{X \vdash C \quad \varnothing_m \vdash D}{X \vdash C \otimes D}\,(\otimes'' r) \qquad \frac{\varnothing_m \vdash C \quad X \vdash D}{X \vdash C \otimes D}\,(\otimes''' r)$$

By "where the antecedent of every sequent is read as a *bunch*$^{*\mathsf{r}}$ we mean that the rule instances of $\mathbf{LBI}^{*\mathsf{r}}$ are precisely those obtained by applying *bunch*$^{*\mathsf{r}}$ to the antecedents of the premise(s) and conclusions of **LBI** rule instances. The reason we have introduced the new rules $(\multimap'\mathrm{r})$, $(\to'\mathrm{r})$, $(\otimes'\mathrm{r})$, $(\otimes''\mathrm{r})$ and $(\otimes'''\mathrm{r})$ is because the rules are derivable in **LBI** but not in $\mathbf{LBI}^{*\mathsf{r}}$ minus the new rules.

Example 2. Every rule instance of $\mathbf{LBI}^{*\mathsf{r}}$ can be obtained from some rule instance of $\mathbf{LBI}^*$ by absorbing all identity structure constants i.e. substructures $\varnothing_m, X$ is transformed to X and $\varnothing_a; X$ is transformed to X and then contracting substructures $X; X$ to X. Indeed, infinitely many distinct rule instances of $\mathbf{LBI}^*$ correspond to the same rule instance in $\mathbf{LBI}^{*r}$. So, for example, the rule instance of $(\to\mathrm{l})$ first row left in $\mathbf{LBI}^{*\mathsf{r}}$ can be viewed as having been obtained by either of the rule instances of $(\to l)$ in $\mathbf{LBI}^*$ first row right and second row.

$$\boxed{\dfrac{Y; C \to D \vdash C \qquad \Gamma[D; C \to D] \vdash A}{\Gamma[Y; C \to D] \vdash A}} \quad \dfrac{Y; C \to D \vdash C \qquad \Gamma[D; D; C \to D] \vdash A}{\Gamma[D; Y; C \to D; C \to D] \vdash A}$$

$$\dfrac{Y; ((Y; \varnothing_a), \varnothing_m); C \to D; C \to D \vdash C \qquad \Gamma[D; C \to D] \vdash A}{\Gamma[Y; ((Y; \varnothing_a), \varnothing_m); C \to D; C \to D] \vdash A}$$

□

Lemma 5. *(i) If* $X \vdash A$ *is* $\boldsymbol{LBI}^*$*-derivable then* $X^{\mathsf{r}} \vdash A$ *is* $\boldsymbol{LBI}^{*\mathsf{r}}$*-derivable. (ii) If* $Y \vdash A$ *is* $\boldsymbol{LBI}^{*\mathsf{r}}$*-derivable and* $X^{\mathsf{r}} = Y$*, then* $X \vdash A$ *is* $\boldsymbol{LBI}^*$*-derivable.*

Proof. (i) Induction on the height of the derivation of $X \vdash A$ in $\mathbf{LBI}^*$. We illustrate with some cases. Suppose the derivation concludes with $(\to\mathrm{l})$ as below left. Let D^k denote $D; \ldots; D$ (k times). Without loss of generality, suppose that $Y \neq Y'; D$ and $V \neq V'; D$ and also that the parent of $V; D^{l+1}$ in $\Gamma[V; D^l; D]$—if it has a parent—is a comma. In the following, a dashed line indicates an argument by the induction hypothesis IH or by properties of the r-function.

$$\dfrac{Y; D^k \vdash C \qquad \Gamma[V; D^{l+1}] \vdash A}{\Gamma[Y; D^k; V; D^l; C \to D] \vdash A}\,(\to l) \qquad \dfrac{\dfrac{Y; D^k \vdash C}{(Y; D^k)^{\mathsf{r}} \vdash C}\,\text{IH} \qquad \dfrac{\Gamma[V; D^{l+1}] \vdash A}{\Gamma^{\mathsf{r}}[V^{\mathsf{r}}; D] \vdash A}\,\text{IH}}{\dfrac{\Gamma^{\mathsf{r}}[(Y; D^k)^{\mathsf{r}}; V^{\mathsf{r}}; C \to D)^{\mathsf{r}}] \vdash A}{\Gamma^{\mathsf{r}}[(Y; V; C \to D; D^k)^{\mathsf{r}}] \vdash A}}\,(\to\mathrm{l})$$

We need to derive in $\mathbf{LBI}^{*\mathsf{r}}$:

$$\begin{cases} \Gamma^{\mathsf{r}}[(Y; V; C \to D)^{\mathsf{r}}] \vdash A & \text{if } k + l = 0 \\ \Gamma^{\mathsf{r}}[(Y; V; C \to D)^{\mathsf{r}}; D] \vdash A & \text{if } k + l > 0 \end{cases} \qquad \begin{cases} \Gamma^{\mathsf{r}}[(Y; V; C \to D)^{\mathsf{r}}] \vdash A & \text{if } k = 0 \\ \Gamma^{\mathsf{r}}[(Y; V; C \to D)^{\mathsf{r}}; D] \vdash A & \text{if } k > 0 \end{cases}$$

If $k = 0$ and $l > 0$ then apply (weak) to $\Gamma^{\mathsf{r}}[(Y; V; C \to D)^{\mathsf{r}}] \vdash A$ to obtain $\Gamma^{\mathsf{r}}[(Y; V; C \to D)^{\mathsf{r}}; D] \vdash A$. Otherwise above right is the required derivation.

Meanwhile if the last rule is (ctr), or $(\varnothing_m\mathrm{l})$ or $(\varnothing_a\mathrm{l})$ in either direction, the induction hypothesis applied to the premise is precisely the required derivation.

(ii) Every rule instance of $\mathbf{LBI}^{*\mathsf{r}}$ can be obtained from some rule instance of $\mathbf{LBI}^*$ by absorbing all identity structure constants and then contractions $X; X \mapsto X$. Because the rules $(\varnothing_m\mathrm{l})$, $(\varnothing_a\mathrm{l})$ and (ctr) witnessing these transformations are in $\mathbf{LBI}^*$, it follows that $Y \vdash A$ is derivable in $\mathbf{LBI}^*$. By assumption, $X^{\mathsf{r}} = Y$ and hence by Lem. 4 we have that $X \vdash A$ is $\mathbf{LBI}^*$-derivable. □

Definition 10 (size). *If α is $\mathbf{1}, \top, \bot$ or a propositional variable then the size $|\alpha| = 1$. If α is $C \heartsuit D$ where $\heartsuit \in \{\wedge, \vee, \to, \otimes, \multimap\}$ then $|\alpha| = |C| + |D| + 1$. Extend to a bunch$^{*\mathsf{r}}$ as follows: $|\varnothing_a| = |\varnothing_m| = 0$. Also*

$$|U_1, \ldots, U_{n+2}| = \sum_{i=1}^{n+2} |U_i| + n + 1 \quad U_i \textit{ not multiplicative}$$

$$|U_1; \ldots; U_{n+2}| = \max\{|U_i|\}_{i=1}^{n+2} \quad U_i \textit{ not additive}$$

In words, the size of a multiplicative bunch$^{*\mathsf{r}}$ is the sum of the sizes of its (necessarily non-multiplicative and non-$\varnothing_m$) bunch$^{*\mathsf{r}}$ children plus the number of children minus one. The size of an additive bunch$^{*\mathsf{r}}$ is the maximum of the sizes of its (necessarily non-additive, non-$\varnothing_a$, non-duplicative) bunch$^{*\mathsf{r}}$ children.

5 Main result

Definition 11 (critical pair). *A* critical pair *(denoted $\{U\}\{V\}$) is an ordered pair where U and V are structures. The set $\mathsf{CP}(X \vdash A)$ of critical pairs of $X \vdash A$ is the smallest set satisfying the following:*

(i) $\{X\}\{A\} \in \mathsf{CP}(X \vdash A)$

(ii) $\{B\}\{B\} \in \mathsf{CP}(X \vdash A)$ *for every formula* B *in* $X \vdash A$
(iii) Antecedent-critical pair: $\{U\}\{B\} \in \mathsf{CP}(X \vdash A)$ *if* $U; B$ *occurs in* X*.*

A critical pair derived by (iii) is called an *antecedent* critical pair because both elements of the pair *necessarily* occur in the antecedent, unlike in (i) and (ii). Also observe that in (iii) it is the case that U and B occur in X as distinct structures. In contrast, in (ii): a single occurrence of a formula is used twice.

Definition 12 (critical pair size). *The size* $|\{U\}\{V\}|$ *of a critical pair* $\{U\}\{V\}$ *is* $|U| + |V|$*.*

Since a bunch$^{*\mathsf{r}}$ is a finite tree, the set of critical pairs of a sequent is finite.

Definition 13 (weight). *The weight* $\mathsf{w}(X \vdash A)$ *of a bunch*$^{*\mathsf{r}}$ *sequent* $X \vdash A$ *is the maximum of the sizes of the critical pairs in* $\mathsf{CP}(X \vdash A)$*.*

Lemma 6. *For every rule instance in* $\boldsymbol{LBI}^{*\mathsf{r}}$*, the weight of each premise is* $\leq$ *the weight of the conclusion.*

Proof. It suffices to show that for every critical pair in a premise, there is a critical pair in the conclusion of greater size. For each rule in $\mathbf{LBI}^{*\mathsf{r}}$:

Consider the critical pair in the premise of type (i). By inspection of each rule, the critical pair of type (i) in the conclusion greater or equal weight.

Consider a critical pair in the premise of type (ii)—i.e. $\{B\}\{B\}$ for some B in the premise. By the subformula property there must be some formula B' occurring in the conclusion such that $|B| \leq |B'|$ and thus $|\{B\}\{B\}| \leq |\{B'\}\{B'\}|$.

Consider a critical pair in the premise of type (iii)—i.e. a critical pair $\{U\}\{B\}$ such that $U; B$ occurs in the premise antecedent. If $U; B$ also occurs in the conclusion of the rule then we are done. The only rules where this might not be the case are the rules where the premise contains a semicolon-separated formula that was not present in the conclusion. Let us consider the possible cases.

Notation. In the following, the critical pair under consideration is $\{U\}\{B\}$. A pair of overbraces over the premise may be used to mark the U and B. The critical pair of greater size in the conclusion is marked by a pair of underbraces, or given in the text when it depends on the relative sizes of the substructures in the premise.

(→r) Here are the possibilities for the critical pair $\{U\}\{B\}$ in the premise.

$$\frac{X; \overbrace{U}; B \vdash D}{\underbrace{X; U} \vdash \underbrace{B \to D}} (\to\mathrm{r}) \qquad \frac{X; \overbrace{U'; C}^{U}; B \vdash D}{X; U'; B \vdash C \to D} (\to\mathrm{r})$$

For example, above left, $\{X; U\}\{B \to D\}$ is a critical pair in the conclusion (of type (i)). Moreover $|\{X; U\}\{B \to D\}| \geq |U| + |B| = |\{U\}\{B\}|$. Above right, the critical pair in the conclusion that we should choose depends on the relative sizes of U' and C. In particular, if $|U'| > |C|$ then $|\{U\}\{B\}| = |U'| + |B|$ so choose the critical pair $\{U'\}\{B\}$ in the conclusion. Else $|C| \geq |U'|$ and $|\{U\}\{B\}| = |C| + |B|$, so choose the critical pair $\{X; U'; B\}\{C \to D\}$ in the conclusion.

($\vee$l) First row left: if $|U'| > |B \vee D|$ then choose $\{U'\}\{B \vee D\}$, else choose $\{B \vee D\}\{B \vee D\}$. First row right: if $|U'| > |C|$ then choose $\{U'\}\{C \vee D\}$, else choose $\{C \vee D\}\{C \vee D\}$. Second row: choose $\{U\}\{B \vee D\}$ as indicated by the underbraces. The other cases are variations of the above.

$$\frac{\Gamma[\overbrace{U'; B \vee D}^{U}; B] \vdash A \qquad \Gamma[U'; B \vee D; D] \vdash A}{\Gamma[U'; B \vee D] \vdash A} \qquad \frac{\Gamma[\overbrace{U'; C}^{U}; \overbrace{C \vee D}^{B}] \vdash A \qquad \Gamma[U'; C \vee D; D] \vdash A}{\Gamma[U'; C \vee D] \vdash A}$$

$$\frac{\Gamma[U; B; B \vee D] \vdash A \qquad \Gamma[U; D; B \vee D] \vdash A}{\Gamma[\underbrace{U}; \underbrace{B \vee D}] \vdash A}\ (\vee\mathrm{l})$$

($\wedge$l) Below left: choose $\{U\}\{B \wedge D\}$ as indicated by the braces. Below center if $|U'| > |C|$ then choose $\{U'\}\{C \wedge D\}$, else choose $\{C \wedge D\}\{C \wedge D\}$. Below right, if $|U'| > |B \wedge D|$ then choose $\{U'\}\{B \wedge D\}$, else choose $\{B \wedge D\}\{B \wedge D\}$.

$$\frac{\Gamma[U; B; D; B \wedge D] \vdash A}{\Gamma[\underbrace{U}; \underbrace{B \wedge D}] \vdash A}\ (\wedge\mathrm{l}) \qquad \frac{\Gamma[\overbrace{U'; C}^{U}; D; \overbrace{C \wedge D}^{B}] \vdash A}{\Gamma[U'; C \wedge D] \vdash A}\ (\wedge\mathrm{l}) \qquad \frac{\Gamma[\overbrace{U'; B \wedge D}^{U}; B; D] \vdash A}{\Gamma[U'; B \wedge D] \vdash A}\ (\wedge\mathrm{l})$$

($\to^{\mathsf{r}}$l) The non-trivial case is given below. Observe that any type (iii) critical pair in the left premise must also be a critical pair in the conclusion.

$$\frac{Y; C \to D \vdash C \qquad \Gamma[\overbrace{C \to D}^{U}; \overbrace{D\ }^{B}] \vdash A}{\Gamma[Y; \underbrace{\underbrace{C \to D}}] \vdash A}\ (\to^{\mathsf{r}}l)$$

(weak) It suffices to observe that any type (iii) critical pair in the premise must also be a critical pair in the conclusion. □

Definition 14 (height of bunch$^{*\mathsf{r}}$). *The* height $\mathsf{h}(X)$ *of a bunch*$^{*\mathsf{r}}$ X *is the number of nodes along its longest branch.*

Lemma 7. *Let X be a bunch*$^{*\mathsf{r}}$*. For $n \geq 0$: if $\mathsf{h}(X) \geq 2n+1$ then $|X| \geq n$.*

Proof. Since $|X| \geq 0$, the result holds trivially when $n = 0$. Next suppose that $n \geq 1$. If $X = U_1; \ldots; U_{k+2}$ then $\mathsf{h}(U_1) \geq 2n$ and $U_1 = V_1, \ldots, V_{l+2}$ and $\mathsf{h}(V_1) \geq 2n-1 = 2(n-1)+1$ (without loss of generality taking $\mathsf{h}(U_1) \geq \mathsf{h}(U_i)$ and $\mathsf{h}(V_1) \geq \mathsf{h}(V_i)$). From the induction hypothesis we have that $|V_1| \geq n-1$. Now $|X| \geq |U_1| \geq 1 + |V_1| = n$. If $X = U_1, \ldots, U_{k+2}$ then $\mathsf{h}(U_1) \geq 2n$ and $U_1 = V_1; \ldots; V_{l+2}$ and $\mathsf{h}(V_1) \geq 2n-1 = 2(n-1)+1$ (without loss of generality taking $\mathsf{h}(U_1) \geq \mathsf{h}(U_i)$ and $\mathsf{h}(V_1) \geq \mathsf{h}(V_i)$). From the induction hypothesis we have that $|V_1| \geq n-1$. Now $|X| \geq 1 + |U_1| \geq 1 + |V_1| = n$. □

Backward proof search on a bunch$^{*\mathsf{r}}$ sequent s_0 is the repeated application of **LBI**$^{*\mathsf{r}}$ rules backwards (i.e. from conclusion to the premises) starting with s_0.

In this way a 'candidate tree' with sequent-labelled nodes is obtained with s_0 at the root. Since there is typically more than one choice of rule to apply, many different candidate trees with root s_0 may be obtained via backward proof search.

Lemma 8. *Let δ be a candidate tree of $\boldsymbol{LBI}^{*\mathsf{r}}$ starting from the bunch$^{*\mathsf{r}}$ sequent s_0. Then the set of bunch$^{*\mathsf{r}}$ sequents that can appear in δ is finite and computable from s_0.*

Proof. Let $\mathsf{sf}(s_0)$ denote the set of subformulae in s_0. Clearly $|\mathsf{sf}(s_0)| \leq 2^{\sum s_0}$ where $\sum s_0$ denotes the sum of the sizes of all formulae in s_0.

Next define the set $\Omega(h)$ of bunch$^{*\mathsf{r}}$ containing only formulae from $\mathsf{sf}(s_0) \cup \{\varnothing_m, \varnothing_a\}$ whose size is bounded by $\mathsf{w}(s_0)$ and whose height is bounded by h.

$$\Omega(h) = \{X \text{ is a bunch}^{*\mathsf{r}} \mid A \in \mathsf{fm} \text{ in } X \text{ implies } A \in \mathsf{sf}(s_0),\ |X| \leq \mathsf{w}(s_0) \text{ and } \mathsf{h}(X) \leq h\}$$

Notice that $\Omega(h) \subset \Omega(h')$ for $h < h'$. First we argue by induction that the number $|\Omega(h)|$ of elements in $\Omega(h)$ is finite and depends only on s_0. For the base case, let us compute $|\Omega(1)|$. If X has height 1 then $X \in \mathsf{Fm}^{\varnothing}$. Then there are only $|\mathsf{sf}(s_0)| + 2$ possibilities for X.

Inductive case. Suppose that $X \in \Omega(k{+}1)\backslash\Omega(k)$. Then X must have the form (i) $U_1, \ldots, U_{n+2}$ (U_i is not multiplicative) or (ii) $U_1; \ldots; U_{n+2}$ (U_i is not additive) where each $U_i \in \Omega(k)$ $(1 \leq i \leq n+2)$. In case (i) we must have $n+1 \leq \mathsf{w}(s_0)$ since otherwise $|X| \geq n+1 > \mathsf{w}(s_0)$ which would contradict $X \in \Omega(k+1)$. Moreover there are at most $|\Omega(k)| < \infty$ choices for each element of the comma-separated list, where this value depends only on s_0 by the induction hypothesis. It follows that the possibilities for X are finite and computable from s_0. In case (ii), because X is a bunch$^{*\mathsf{r}}$ structure we have $U_i = U_j$ iff $i = j$. So the possibilities for X are limited to the elements in $\mathcal{P}(\Omega(k))\backslash\emptyset$ and thus the number of possibilities is bounded by $2^{|\Omega(k)|}$ ($\mathcal{P}$ is the powerset operator).

We have shown $\Omega(h)$ is finite for every h depending only on s_0. Moreover for a bunch$^{*\mathsf{r}}$ sequent $X \vdash A$ appearing in the backward proof search, it must be the case that $\mathsf{h}(X) < 2(\mathsf{w}(s_0)+1)+1$ for otherwise we would have $|X| \geq \mathsf{w}(s_0)+1$ (Lem. 7) which is impossible by Lem. 6. It follows that $X \in \Omega(2\mathsf{w}(s_0)+3)$. Moreover we have that $A \in \mathsf{sf}(s_0)$. Therefore the number of possible sequents that may appear in a candidate tree is bounded by $|\Omega(2\mathsf{w}(s_0)+3)| \cdot |\mathsf{sf}(s_0)|$. □

Corollary 1. ***BI*** *is decidable.*

Proof. By Lem. 2 and 5 it suffices to show that $\mathbf{LBI}^{*\mathsf{r}}$ is decidable. Given bunch$^{*\mathsf{r}}$ sequent s_0, generate all candidate trees, terminating a branch if a sequent is encountered twice on the same branch (loop check). We disregard candidate trees with such repetitions because then there must be a smaller derivation of s_0. From Lem. 8 the candidate trees are finite and can be effectively generated. If one of these trees is a derivation then s_0 is derivable, otherwise it is not. □

6 Conclusion

The bunch$^{*\mathsf{r}}$ normal form was obtained to facilitate a concise definition of critical pairs and height of a bunch. In particular, the arguments could have been made directly on the bunches by consideration of 'connected regions' of commas/semicolons in the bunch. E.g. the height of a bunch would be the maximum number of transitions between comma and semicolon regions counting from the root. Recently *intuitionistic layered graph logic* [3] has been proposed as a logic for reasoning about layers e.g. the infrastructure and social layer in a transport network, the relationship between a security policy and the system architecture. This logic is obtained from **BI** by replacing **MILL** with its non-commutative non-associative counterpart. It is plausible that a similar argument to the one given here could be used to obtain decidability (this time argue on bunches directly since the reduction to bunch$^{*\mathsf{r}}$ is not possible). Another extension would be to add the contraction rule for $\otimes$ to **BI**. We are unaware yet of a resource-interpretation for this logic, but it would be interesting to see if Kripke's argument [9] for bounding $\otimes$-contraction in $\mathbf{FL}_{ec}$ can be incorporated here.

Notice that a similar argument could *not* even be attempted for **BBI** (the logic is in any case known [10] to be undecidable) because no cutfree sequent calculus for **BBI** is known.

Lem. 8 implies that backward proof search in **BI** terminates. Thus we can use the calculus $\mathbf{LBI}^{*\mathsf{r}}$ for backward proof search/automated proving. The result also yields an upper bound for the complexity of **BI**. Given that a syntactic proof of decidability has already proved so vexing, we defer the explicit calculation of bounds to future work.

The literature contains the presentations of many different logics via cutfree sequent calculi (and also using notions of analyticity weaker-than-cutfree but seemingly powerful nonetheless), and also via generalisations of the sequent formalism such as hypersequent and nested sequent calculi. And yet it is often unclear how to make use of these cutfree calculi to obtain a decision procedure due to difficulties in bounding the backward proof search. This in turn is due to the problematic interactions between certain rules in the calculus. For example, for the fuzzy logics **MTL** (decidable via semantics with no known complexity bound) and **UL** (decidability problem open), it is the interaction of the communication and external contraction rules in the hypersequent calculus which causes difficulty [11]. Some techniques to control backward proof search include loop check, the simplifications obtained from the careful elimination of problematic rules such as the contraction rule, and the use of novel parameters (e.g. [16]) which capture specific aspects of the calculus. The weight measure in this paper belongs to the latter category. We believe that it is imperative to build a toolkit of methods to tackle the decision problem in the various calculi. Not only will this lead to new backward proof search procedures and complexity bounds, such applications will provide tangible evidence that an analytic calculus is a fundamentally different object to non-analytic presentations of the logic (e.g. Hilbert calculus). We view this paper as work in this direction.

References

1. Armelín, P.A., Pym, D.J.: Bunched Logic Programming, pp. 289–304. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
2. Chvalovský, K., Horcík, R.: Full lambek calculus with contraction is undecidable. To appear in J. Symb. Log.
3. Docherty, S., Pym, D.J.: Intuitionistic layered graph logic. In: Automated Reasoning: 8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, Proceedings. pp. 469–486. Springer International Publishing (2016)
4. Dyckhoff, R.: Contraction-free sequent calculi for intuitionistic logic. J. Symbolic Logic 57(3), 795–807 (1992)
5. Galmiche, D., Méry, D., Pym, D.: The semantics of BI and resource tableaux. Math. Structures Comput. Sci. 15(6), 1033–1088 (2005)
6. Gentzen, G.: The collected papers of Gerhard Gentzen. Edited by M. E. Szabo. Studies in Logic and the Foundations of Mathematics, North-Holland Publishing Co., Amsterdam (1969)
7. Ishtiaq, S.S., O'Hearn, P.W.: Bi as an assertion language for mutable data structures. In: Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. pp. 14–26. POPL '01, New York, NY, USA (2001)
8. Kamide, N.: Linear exponentials as resource operators: A decidable first-order linear logic with bounded exponentials. In: Logics in Artificial Intelligence: 11th European Conference, JELIA 2008, Dresden, Germany. pp. 245–257 (2008)
9. Kripke, S.: The problem of entailment (abstract). J. Symbolic Logic 24, 324 (1959)
10. Larchey-Wendling, D., Galmiche, D.: The undecidability of Boolean BI through phase semantics. In: 25th Annual IEEE Symposium on Logic in Computer Science LICS 2010, pp. 140–149. IEEE Computer Soc., Los Alamitos, CA (2010)
11. Metcalfe, G., Olivetti, N., Gabbay, D.: Proof Theory for Fuzzy Logics, Springer Series in Applied Logic, vol. 39. Springer (2009)
12. O'Hearn, P.W., Pym, D.J.: The logic of bunched implications. Bull. Symbolic Logic 5(2), 215–244 (1999)
13. Pym, D.J.: The semantics and proof theory of the logic of bunched implications, Applied Logic Series, vol. 26. Kluwer Academic Publishers, Dordrecht (2002), with a foreword by Dov M. Gabbay
14. Pym, D.J., O'Hearn, P.W., Yang, H.: Possible worlds and resources: the semantics of BI. Theor. Comput. Sci. 315(1), 257–305 (2004)
15. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence, Prentice Hall, 3rd edn. (2010)
16. Valentini, S.: The modal logic of provability: cut-elimination. J. Philos. Logic 12(4), 471–476 (1983)